Circuits et tris topologiques dans les graphes orientés

1 Présentation générale

Nous avons vu en cours une notion de *circuit* dans un graphe orienté. La recherche de circuit dans un graphe est un thème algorithmique assez facile. Il devient plus intéressant lorsqu'on pose la question: comment prouver qu'un graphe orienté n'a pas de circuit?
Nous allons explorer de manière aprofondie cette question dans ce TD. Il mélangera des questions théoriques, et des implémentations en Python. Pensez bien à tester tout ce que vous implémentez.

2 Graphes orientés sans circuits

Le sujet de la certification de l'absence de circuit est plus facile à traîter sur le plan algorithmique dans les graphes sans circuit. Nous allons donc nous limiter dans un premier temps à ce contexte; nous passerons dans un second temps au cas général.

2.1 Puits et circuits

Afin de travailler sur les circuits et les tris topologiques, nous allons utiliser la notion de puits, et la recherche de puits en particulier.
Voici un algorithme qui trouve un puits dans un graphe sans circuit:

Trouve_Puits($\vec{G} = (V, A)$, $s \in V$):
– Tant que $|N^+(s)| \neq 0$, faire:
  * Soit $x \in N^+(s)$;
  * $s \leftarrow x$;
– **Retourner** s.

1. Ecrire une fonction Python qui implémente `Trouve_Puits()`.

2. Démontrer que l'algorithme `Trouve_Puits()` s'exécute toujours correctement; finit par s'arrêter quelle que soit son entrée; et que lorsqu'il s'arrête, il retourne bien un puits du graphe.

3. Un circuit peut-il passer par un puits?
Cette dernière question a une allure un peu triviale. Il n'y a pas beaucoup plus de profondeur au sujet de l'absence de circuit dans un graphe orienté. En effet, lorsque vous aurez fini ce TD, vous aurez démontré que dans un graphe orienté quelconque, il y a un circuit, si et seulement si il reste des sommets dans le graphe une fois qu'on lui a retiré successivement tous ses puits.

2.2 Tri topologique: théorie

Voici la définition de tri topologique.
On dit qu'une fonction $\phi : V \to \mathcal{Z}$ est un tri topologique du graphe $\vec{G} = (V, A)$ si:

$$\forall \vec{xy} \in A, \quad \phi(x) < \phi(y)$$

.

4. Donnez un exemple de graphe, avec un exemple de tri topologique dans ce graphe.

5. Démontrez le théorème suivant: "Il ne peut y avoir dans un graphe orienté à la fois un circuit et un tri topologique"

6. Démontrez le fait suivant. Si $\vec{G} = (V, A)$ est un graphe orienté, p est un puits de $\vec{G}$, et $\vec{G}[V \setminus \{p\}]$ admet un tri topologique, alors $\vec{G}$ admet aussi un tri topologique.

2.3 Tri topologique: pratique

7. Ecrire une fonction en Python qui calcule un tri topologique d'un graphe sans circuit.

8. Démontrez le fait suivant: Si $\vec{G}$ est un graphe orienté sans circuit, alors $\vec{G}$ a un tri topologique.

9. Exhibez un certificat du NON pour le problème de la recherche d'un circuit dans un graphe orienté.

3 Graphes orientés quelconques

10. Ecrire une foncion Python `Circuit_ou_Puits()` qui trouve soit un circuit, soit un puits, dans un graphe orienté quelconque. Elle prendra en entier un graphe orienté, et un sommet de ce graphe.

11. Ecrire une fonction Python `Circuit_ou_Tri_topologique()` qui trouve soit un circuit, soit un tri topologique dans un graphe orienté.

12. Démontrez que dans un graphe orienté, il y a toujours soit (au moins) un circuit, soit (au moins) un tri topologique, les deux possibilités étant exclusives l'une de l'autre.

13. Démontrez que pour un graphe orienté $\vec{G}$, $\vec{G}$ a un circuit si et seulement si, en retirant successivement les puits dans $\vec{G}$, il reste des sommets.
